Task: KOR
Necklace

Olimpiada Informatyczna

XXIII OI, Stage I. Source file `kor.*` Available memory: 256 MB. *19.10–16.11.2015*

Bytina has n pairwise different beads, numbered from 1 to n, each of a certain known value.

She would like to make a necklace from some of her beads, but has hard time selecting the particular subset of beads and their arrangement. To simplify the choice, she decided to ignore the bead arrangement completely, i.e., deem two necklaces different only if the sets of their constituent beads differ. To make her choice even simpler, she now wants to introduce an order among all possible necklaces.

The total value of the beads is Bytina's single most important criterion. Thus, the higher this value, the later a necklace should appear in the order. However, should there be several necklaces of the same value, these should be ordered lexicographically with respect to the sequences of increasing values of their constituent beads*.

For example, consider a scenario with four beads of values (in the bead numbers order) 3, 7, 4 and 3. 16 different necklaces can be made of these beads, listed below in Bytina's order.

Necklace no.	Value of chosen beads	Total Value of chosen beads	Numbers of chosen beads
1	*none*	0	*none*
2	3	3	1
3	3	3	4
4	4	4	3
5	3 3	6	1 4
6	3 4	7	1 3
7	7	7	2
8	4 3	7	3 4
9	3 7	10	1 2
10	3 4 3	10	1 3 4
11	7 3	10	2 4
12	7 4	11	2 3
13	3 7 3	13	1 2 4
14	3 7 4	14	1 2 3
15	7 4 3	14	2 3 4
16	3 7 4 3	17	1 2 3 4

Finally, Bytina has made up her mind! She wants the k-th necklace in her order. Tell her which one that is!

Input

In the first line of the standard input, there are two positive integers n and k, separated by a single space, specifying the number of beads and the desired necklace number in Bytina's order. In the second line of the input, there is a sequence of n positive integers $a_1, a_2, \ldots, a_n$, separated by single spaces, which specify the successive bead's values.

You may assume that Bytina made no mistake and there are indeed at least k different necklaces.

Output

In the first line of the standard output, a single integer should be printed: the total value of the beads in the desired necklace. In the second line of output, the increasing sequence of values of the constituent beads should be printed, separated by single spaces.

*A bead sequence $i_1, \ldots, i_p$ is lexicographically smaller than the sequence $j_1, \ldots, j_q$ if either the first sequence is a prefix of the second one (i.e., $p < q$, $i_1 = j_1$, $\ldots$, $i_p = j_p$), or at the first position where these two sequences differ, the first sequence has a smaller element than the second one (i.e., there exists $u \in \{1, \ldots, \min(p, q)\}$ such that $i_1 = j_1$, $\ldots$, $i_{u-1} = j_{u-1}$ and $i_u < j_u$).

Example

For the input data:	the correct result is:
`4 10` `3 7 4 3`	`10` `1 3 4`

Sample grading tests:

`1ocen:` $n = 10$, all the beads have value 1,

`2ocen:` $n = 9$, the bead values are successive powers of two,

`3ocen:` $n = 11$, there is a single bead worth 1 and 10 beads worth 10^9 each, and the desired necklace uses all 11 beads,

`4ocen:` $n = 1\,000\,000$, $k = 10$, the value of bead no. i is i.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several test groups.

Each subset satisfies the conditions $n, k \leq 1\,000\,000$ and $a_i \leq 10^9$.

If the answer to a given test is incorrect but the first line of the output (the total value of the beads in the necklace) is correct, then half the score for the test is awarded (appropriately scaled if half the time limit was exceeded). Note that this applies not only when the second line of output is incorrect, but also when there is no such line, or even when more than two lines are output.

Subset	Property	Score
1	$n \leq 20$, $k \leq 500\,000$	8
2	$n \leq 60$, $k \leq 50\,000$	12
3	$n \leq 3\,000$, $n \cdot k \leq 10^6$, $a_i \leq 100$	14
4	$n \cdot k \leq 10^6$	16
5	$n \cdot k \leq 10^7$	20
6	none	30